doi: 10.20532/cit.2024.1005825

High-Frequency Quantitative Trading of Digital Currencies Based on Fusion of Deep Reinforcement Learning Models with Evolutionary Strategies

Yijun He[1, 2], Bo Xu[1, 2] and Xinpu Su[1, 2]

[1]School of Information Science, Guangdong University of Finance and Economics, China
[2]Guangdong Intelligent Business Engineering Technology Research Center, China

High-frequency quantitative trading in the emerging digital currency market poses unique challenges due to the lack of established methods for extracting trading information. This paper proposes a deep evolutionary reinforcement learning (DERL) model that combines deep reinforcement learning with evolutionary strategies to address these challenges. Reinforcement learning is applied to data cleaning and factor extraction from a high-frequency, microscopic viewpoint to quantitatively explain the supply and demand imbalance and to create trading strategies. In order to determine whether the algorithm can successfully extract the significant hidden features in the factors when faced with large and complex high-frequency factors, this paper trains the agent in reinforcement learning using three different learning algorithms, including Q-learning, evolutionary strategies, and policy gradient. The experimental dataset, which contains data on sharp up, sharp down, and continuous oscillation situations, was chosen to test Bitcoin in January-February, September, and November of 2022. According to the experimental results, the evolutionary strategies algorithm achieved returns of 59.18%, 25.14%, and 22.72%, respectively. The results demonstrate that deep reinforcement learning based on the evolutionary strategies outperforms Q-learning and policy gradient concerning risk resistance and return capability. The proposed approach offers a robust and adaptive solution for high-frequency trading in the digital currency market, contributing to the development of effective quantitative trading strategies.

ACM CCS (2012) Classification: Computing methodologies → Machine learning → Machine learning algorithms → Feature selection

Computing methodologies → Machine learning → Learning paradigms → Reinforcement learning → Sequential decision making

Keywords: reinforcement learning, Bitcoin, quantitative trading, evolutionary strategies

1. Introduction

Bitcoin, since its inception in 2009, has pioneered the field of decentralized, peer-to-peer digital currencies, marking a revolutionary shift from traditional financial systems. Its rapid ascension to mainstream recognition, as a credible alternative to conventional currencies, underscores a global appetite for innovative financial instruments. Central to Bitcoin's disruption is its groundbreaking decentralized architecture, leveraging blockchain technology to facilitate secure, efficient transactions between untrusted entities, thereby fostering an unprecedented level of trust and autonomy in each exchange. Amidst this transformative landscape, high-frequency quantitative trading (HFQT) has emerged as a pivotal force shaping the digital currency market. This practice, characterized by algorithm-driven, ultra-fast trade executions, introduces new dimensions of liquidity and efficiency, yet simultaneously poses complex challenges related to market volatility, regulatory oversight, and the need for advanced technological infrastructure. Hence, understanding the implications of HFQT within the context of Bitcoin and broader cryptocurrency ecosystems becomes imperative for navigating the future of digital finance.

In the fast-paced domain of digital currency trading, identifying potent characteristic factors is vital for the formulation of high-frequency quantitative strategies. Conventional approaches, heavily reliant on K-chart patterns, often falter in providing actionable insights. These models, while predicting future closing prices, inadequately address the dynamic complexity of real-time trading environments. They provide only a prediction of the future closing price and do not output specific actions (such as buying, selling, waiting, *etc.*), making them only useful as reference factors and not as direct guides for practice.

To surmount these limitations, this investigation pivots towards deep reinforcement learning (DRL), harnessing its prowess in deciphering intricate patterns from high-frequency, granular data. Specifically, we employ one-minute candlestick data for meticulous data preprocessing and feature extraction, going beyond the superficial analysis of conventional techniques. By integrating deep neural networks (DNN), our methodology delves into the latent structures within transaction volumes and timestamps, unearthing subtle market dynamics often obscured by traditional methodologies.

The core objective of this study is to present the deep evolutionary reinforcement learning (DERL) model, a novel framework designed to transform extracted features into precise trading actions. DERL not only predicts market movements but also formulates a strategic response matrix, directly guiding traders with buy/sell/hold signals informed by a nuanced understanding of supply-demand imbalances. This study bridges the gap between theoretical predictions and practical trading execution, enhancing the efficacy and responsiveness of high-frequency trading strategies in the digital currency sphere. Our contribution lies in demonstrating the superior adaptability and predictive power of DERL, offering a robust framework that promises to redefine the efficacy of high-frequency quantitative trading strategies.

The rest of the paper is structured as follows. The second part is the current status and summary of factor mining and stock selection model methods in China and the rest of the world and proposes research ideas and methods after identifying the research object of this paper as high-frequency factors in the digital currency market. The third part applies a deep reinforcement learning model based on an evolutionary strategy to extract high-frequency factors and describes each part of the model in detail. The fourth part is the experimental method design and results analysis. Finally, the fifth part summarizes the findings of this experiment.

2. Related Research

After years of research, numerous factor mining and stock selection model methods have been proposed for the trading market. Furthermore, with the development of the digital currency market and computer technology, the application of high-frequency quantitative trading and high-frequency factor mining in market investment has become increasingly prevalent.

Han *et al.* [1] divided the investment environment into two main parts: static market state and dynamic portfolio position state. Using this method, they designed a natural portfolio return mechanism based on probabilistic dynamic programming. This mechanism endows the DRL agent with stronger environmental adaptation and interactive intelligence, enabling it to learn more efficiently from environmental feedback and reduce the interaction cost required for exploration, thereby improving the overall performance and robustness of the investment strategy. Mattera and Raffaele [2] introduced a method based on reinforcement learning to estimate high-dimensional covariance matrix for portfolio selection problems. In this paper, two reinforcement learning models were proposed, both of which use neural networks to approximate the optimal strategy, that is, to select the best contraction strength parameters to construct the contraction covariance matrix. Xu *et al.* [3] proposed an end-to-end deep reinforcement learning automated trading algorithm that integrates CNN and LSTM, which can perceive dynamic market conditions of stocks and extract dynamic features, cycle learning dynamic time series patterns, and accumulate final profits through reinforcement learning methods to optimize the investment portfolio.

In the high-frequency and multidimensional digital currency trading environment, reinforcement learning can perform asset allocation tasks more precisely, thereby improving ROI and enhancing risk management capabilities. Liu *et al.* [4] proposed a reinforcement learning-driven asset allocation strategy optimization solution called LSRE-CAAN. The core of this solution is to use the policy gradient (PG) algorithm for optimization calculations and to deeply mine historical trading data to construct the state transition matrix and action value function, thereby achieving dynamic adjustment of the asset portfolio. Duan *et al.* [5] proposed a new deep reinforcement learning algorithm called OASS. In this approach, the trading decision is treated as a sequential decision problem, and the algorithm models the time-dependence of the environment state using multi-layer LSTM networks. Additionally, the algorithm uses a probabilistic dynamic programming algorithm to handle high noise and uncertainty in the data.

For the more complex digital currency market, the proximal policy optimization (PPO) algorithm has gained favor among many scholars in academia due to its high stability, strong interpretability, wide applicability, and considerable practical prospects compared to other algorithms. Li *et al.* [6] proposed an automatic high-frequency Bitcoin trading framework based on PPO algorithm, a deep reinforcement learning algorithm. The paper used LSTM as a foundation to build the policy function. This paper showed the possibility of building a single cryptocurrency trading strategy based on deep learning and provided some implications for future research. Chung *et al.* [7] explained the market making reinforcement learning agent based on the order stacking framework. This paper uses PPO algorithm to update strategy function and value function, so that agents can adapt and improve in different market environments. Chen and Guo [8] designed a trading strategy applicable to China's futures market by using PPO algorithm, which could realize the end-to-end decision-making process from data to trading behavior from the historical trading price and volume data of rebar futures. The results show that the proposed strategy has a higher proportion in the profitability test period and is more adaptable to different market conditions.

In order to solve the problem of risk control in quantitative trading in the face of asymmetric risks, Alameer and Shehri [9] first introduced conditional value at risk (CVaR) as a performance indicator into quantitative trading of direct reinforcement learning (DRL) and utilized the convexity of CVaR to ensure the convexity of the problem. However, there are still some areas for further improvement when dealing with high dimensional data. Therefore, in order to improve the efficiency and accuracy of the algorithm. Cui *et al.* [10] proposed a new approach based on conditional value at risk (CVaR) and PPO algorithms for constructing an efficient portfolio in the cryptocurrency market with large tail risks.

Suliman *et al.* [11] adopted the competitive dual deep Q network (dueling DQN). The algorithm can estimate the state value function and the advantage function of each action respectively, thus reducing the problem of overestimation of the Q-value. In the reinforcement learning framework, Wang *et al.* [12] adopted experiential replay and iteration mechanisms to solve complex problems and added dual DQN and dueling DQN structures to optimize the neural network structure. Through the combination of turning point classification and reinforcement learning framework, the method was able to better adapt to market changes and improve trading returns.

At the same time, in the face of complex market environment and nonlinear trading relationship, Huang *et al.* [13], taking crude oil and natural gas futures market as an example, applied two-branch deep Q-network (TBDQN) to crude oil and natural gas futures trading to automatically generate stable profits and robust trading signals. This paper comprehensively considered the short-term and long-term returns, as well as the correlation between various factors and returns, and proposed a correlation-based reward function. At the same time, discount factors were used to adjust the reward value at different time points, so that the reward function can reflect the balance of short-term and long-term returns. Zhao *et al.* [14] proposed a stock prediction method that combines the median absolute deviation (MAD) method and Q-learning model. The MAD method was used to preprocess data and set reward value reasonably to improve the prediction performance and actual effect of the Q-learning method. At the same time, it improved the efficiency and accuracy of stock trading decisions.

In summary, a large number of scholars have discussed the quantitative trading practices of deep reinforcement learning in different financial markets, including stock trading, futures trading, option trading, foreign exchange market and cryptocurrency market, and have classified and compared different types of deep reinforcement learning algorithm models. However, few researchers have applied high-frequency factor mining technology to the digital currency market. Therefore, this paper proposes to use deep reinforcement learning for high-frequency factor extraction. Taking Bitcoin as an example, one-minute K-line data is used as base data in the digital currency market, and data cleaning and optimization of one-minute K-line high-frequency information are selected from the perspectives of high frequency and micro, to carry out factor extraction.

Table 1. Limitation analysis of the algorithm model.

Han *et al.* [1]	Breaking down the environment into static market states and dynamic portfolio position states is a simplified approach that takes into account only changes in asset prices and ignores other factors that affect investment decisions.
Mattera and Raffaele [2]	The paper uses a simple multi-layer perceptron network as a policy approximation, which may not capture the complex features and structure of the high-dimensional covariance matrix, and lacks a mechanism for regularization or dimensionality reduction, which may lead to problems of overfitting or dimensional disaster.
Xu *et al.* [3]	There are many parameters in the model.
Liu *et al.* [4]	This method uses probabilistic dynamic programming algorithm to avoid the sampling process, so it may lead to low training efficiency.
Duan *et al.* [5]	The model can only use historical price information to construct a single asset HFT strategy, without taking into account other factors that may affect the Bitcoin market.
Li *et al.* [6]	The model uses fixed step intervals to determine actions without taking into account the dynamics of the market.
Chung *et al.* [7]	The model uses a shared fully connected layer to process different input features, which may lead to mutual interference or information loss between features.
Chen and Guo [8]	The model uses a single-layer neural network as a strategy function, which may not capture complex patterns in market data.
Alameer and Shehri [9]	The model does not consider the influence of different time windows on the model.
Cui *et al.* [10]	This model can only handle the behavior control task of a single agent and can not adapt to the decision problem of multi-agent cooperation.
Suliman *et al.* [11]	CNN may ignore long-term dependencies on time series, and LSTM may have difficulty dealing with high and non-linear data.
Wang *et al.* [12]	The action space of the model is too simple, with only two actions, without considering more detailed trading decisions or more action options.
Huang *et al.* [13]	The classification branch of the TBDQN model uses a binary cross entropy (BCE) loss function, which can lead to problems with class imbalance.
Zhao *et al.* [14]	When designing deep Q-learning network, the model is not optimized for specific problems.

3. Deep Reinforcement Learning Based on Fused Evolutionary Strategies (DERL) Trading Model

3.1. Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment in order to maximize a reward signal. Unlike supervised and unsupervised learning, which focus on learning from labeled data or discovering patterns in data, reinforcement learning is focused on learning from experience. The agent takes actions in the environment and receives feedback in the form of rewards or penalties. Over time, the agent learns which actions lead to the highest rewards and adjusts its behavior accordingly. The main elements of reinforcement learning are the agent, the environment, actions, and rewards. The goal is to find the optimal policy, or sequence of actions, that maximizes the cumulative reward over time.

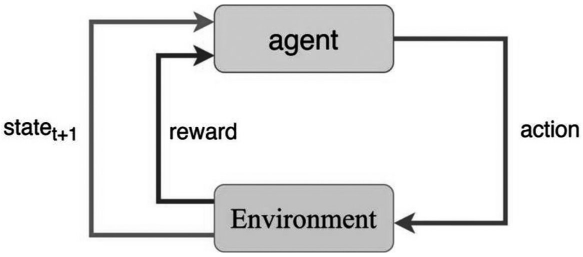

Figure 1. Flow chart of reinforcement learning.

As shown in Figure 1, reinforcement learning is inspired by behaviorist psychology, and the Markov decision process (MDP) is used as a model to construct two main subjects: an intelligent body and the environment, so that the intelligent body observes the state of the environment (State), then makes an action (Action), gets the reward of feedback (Reward). The goal is to give the intelligent body the ability to make the optimal action decision in each state and maximize the future cumulative reward.

Deep reinforcement learning is a combination of reinforcement learning and deep learning, where deep learning is used to handle the perception problem and reinforcement learning is used to handle the decision problem. The traditional reinforcement learning algorithms, like Q-learning and SARSA, use tables to store the value of actions and update the policy based on experience. However, this table-based approach is unsuitable for scenarios with high-dimensional state and action spaces. Deep reinforcement learning overcomes this limitation by using neural networks to solve the decision problem, instead of tables. This allows deep reinforcement learning to handle high-dimensional state spaces and make more informed decisions.

3.2. Deep Evolutionary Reinforcement Learning Trade-selection Algorithms Incorporating Evolutionary Strategies

The deep evolutionary reinforcement learning (DERL) algorithm is an algorithm that combines the evolutionary strategies (ES) algorithm with the neural network approximation function. The reason for combining ES algorithms with deep reinforcement learning models in this paper is that evolutionary algorithms generally have the advantages of better global search capability, good robustness, and parallelism, so combining evolutionary algorithms with deep reinforcement learning can compensate for the lack of exploration capability of deep reinforcement learning environment, poor robustness, and susceptibility to deceptive gradients caused by deceptive rewards. The evolutionary strategy adds the expectation of the cumulative discounted return over time. In this paper, we apply the DERL algorithm model to train an intelligent agent that can master the timing rules of the digital currency market, and then apply it to a quantitative trading system to exploit the agent's timing decision-making ability.

Specifically, the core of this framework integrates a set of random perturbations into the existing policy parameter set via evolutionary strategies. Each perturbation can be viewed as an exploratory leap in the solution space, aimed at broadening the search scope and unearthing potentially high-performance strategies. Subsequently, leveraging these modified parameter configurations, agents are deployed into environments to execute sequences of interactions, with accumulated rewards during this period serving as the key metric for evaluating the adaptability of the policies. Based on this feedback mechanism, natural selection principles are applied to cull out parameter variants that yield superior cumulative returns, thereby achieving an efficient screening and optimization of the policy space. This iterative cycle progresses, with each round dedicated to pushing the boundaries of policy performance until a predetermined convergence criterion is met or specific performance thresholds are satisfied.

Distinguishing itself from traditional gradient descent techniques that incrementally optimize along the negative gradient path, ES actively explores the parameter space through stochastic perturbations of the strategy, effectively avoiding the common issues of gradient vanishing and explosion encountered in high-dimensional deep networks. Moreover, it demonstrates robustness and exploration capability in complex optimization scenarios. Consequently, ES furnishes a potent tool for advancing the optimization of deep learning models in non-gradient or weak-gradient settings.

Figure 2 shows the modules in the DERL network from the perspective of data flow, its components are described in detail below.

1. Environment (Env): OHLCV data is a time series, with obvious trends and seasonality, both of which affect the algorithm's ability to predict the time series [15], so the environment first needs to normalize the data in the current market state, so that the input data is smoothed, thus making the distribution of the data more normal [16].
2. Reward function setting: The simple buy-and-hold strategy is intuitive, but it produces an unstable strategy that often leads to serious losses of capital. Therefore, the risk-adjusted return indicator is chosen as the reward function to better optimize the model. The most common risk-adjusted return metric is the Sharpe ratio. This is a simple ratio of portfolio excess return to volatility, measured over a specific period. In order to maintain a high Sharpe ratio, investments must have high returns and low volatility (risk). It is calculated as:

$$SharpeRatio = \frac{E(R_P - R_f)}{\sigma_P} \tag{1}$$

where $E(R_P)$ is the expected annualized portfolio payoff, R_f is the annualized risk-free rate, and σ_P is the standard deviation of the annualized portfolio payoff.

However, this indicator is flawed for Bitcoin because it penalizes upside volatility (wild price increases). In the virtual currency market, upside volatility is often part of what allows for quick gains. The Sortino ratio is very similar to the Sharpe ratio, except that it only considers down-

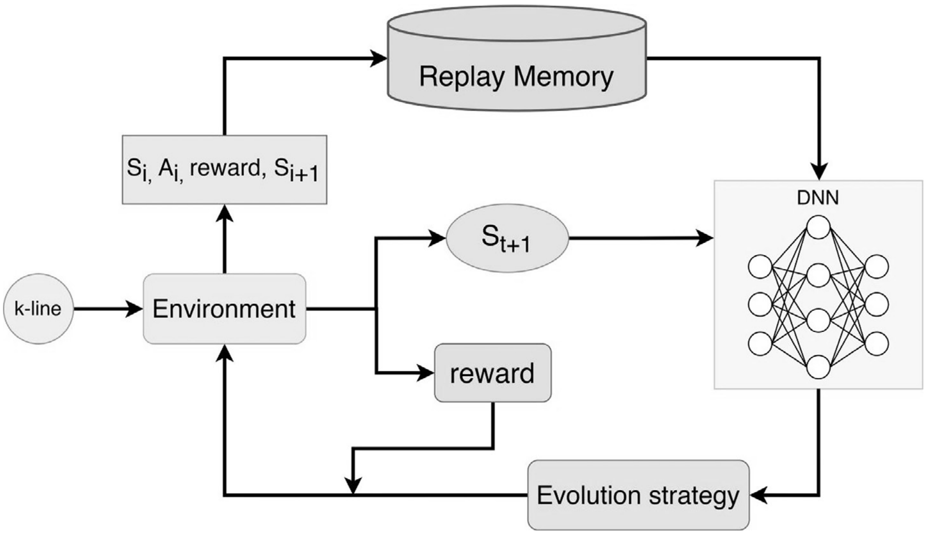

Figure 2. Deep reinforcement learning (DERL) flowchart incorporating evolutionary strategies. DNN stands for deep neural network.

side volatility as risk, not overall volatility. Therefore, this ratio does not affect upside volatility:

$$SortinoRatio = \frac{E(R_P) - MAR}{\sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(R_{PT} - R_f)^2}} \qquad (2)$$

The MAR stands for minimum acceptable return, R_{PT} is a measure of all samples in the return column that are smaller than the risk-free return.

3. Experience Replay Pool (Replay Memory): Setting up an experience replay pool can be a good solution to the two shortcomings of On-policy that generally uses data from one episode for parameter updates, and the data are used up and lost: namely, the data (states) are interrelated, the data do not have the nature of independent identical distribution (i.i.d.), many popular stochastic gradient algorithms tend to have the assumption that the data are independently identically distributed and are quick to forget about rare (sparse) experiences, ignoring the role that these rare experiences may be more useful to update multiple times. Experience pools break their correlation about time by mixing recent experiences and learning multiple times by sampling experiences from the experience pool [17]. Each time the DERL parameters are updated, the data needed is simply a quadratic (s_t, a_t, r_t, s_{t+1}) with the actual strategy π being not very relevant, so the previously collected experimental quaternions can be used iteratively to train the model and update the parameters. For the training data $< s, a, r, s' >$, the individual is able to obtain a training sample for each sequential action performed, so $< s, a, r, s' >$ is put into the experience pool after each sequential action performed. However, since the states are continuous, there must be some correlation between the training samples put in sequence, so that the trained neural network is prone to overfitting. In order to solve this problem, a small amount of training data is randomly selected from the experience pool as a batch, which ensures that the training samples are independently and identically distributed and speeds up the training rate.

4. Evolution strategy: Evolutionary strategies is a gradient-free stochastic optimization algorithm, which is the main idea of doing black-box optimization, *i.e.*, adjusting a normal distribution for search by iterations. The normal distribution of iterations in the evolutionary strategies is generally written as $N(m_t, \sigma_t^2, C_t)$ and contains three parameters: m_t, σ_t^2, C_t. The role played by the parameters of the normal distribution is:

   - m_t is mean: determines the distribution's central location, determines the algorithm's search area.
   - σ_t is the step parameter, which determines the overall variance of the distribution (global variance), determines the size and intensity of the search range in the algorithm.
   - C_t is the covariance matrix, which determines the shape of the distribution, determines the dependencies between variables and the relative scales between search directions in the algorithm.

The core of the ES algorithm design is how to adjust these parameters, especially the step size parameter and the covariance matrix, to achieve the best possible search results. The tuning of these parameters has a very important impact on the convergence rate of the ES algorithm. In general, the basic idea of ES tuning parameters is to adjust the parameters so that the probability of producing a good solution gradually increases (the probability of searching along the good search direction increases).

The idea of applying evolutionary algorithms to reinforcement learning goes back a long way [18], but due to computational limitations, this attempt only stopped at "tabular" reinforcement learning (Q-learning). Inspired by NES, researchers at OpenAI [19] proposed to use NES as a non-gradient black-box optimizer to find the optimal policy parameters that maximize the return function $F(\theta)$ of the optimal policy parameters θ.The key is to add Gaussian noise to the model parameters and use a likelihood trick to write the gradient of the Gaussian probability density function so that, ultimately, only the noise term remains as a weighted scalar measure of performance.

Suppose the current parameter value is $\hat{\theta}$ (for differentiating the random variable θ), the θ search distribution is designed as an isotropic multivariate Gaussian distribution with mean $\hat{\theta}$ and the covariance matrix $\sigma^2 I$.

$$\theta \sim N\left(\hat{\theta}, \sigma^2 I\right) \text{ equivalent to } \theta = \hat{\theta} + \sigma\varepsilon, \\ \sigma \sim N(0, I) \tag{3}$$

In each generation, we can sample to get many ε_i ($i = 1, ..., n$) and then estimate their fitnesses in parallel. An elegant design approach eliminates the need to share large model parameters. Simply passing random seeds between worker threads is sufficient for the main thread nodes to perform parameter updates. Subsequently, this approach was extended to learn the loss function in an adaptive square trial. To make the performance of the algorithm more robust, OpenAI ES employs virtual BN (a batch normalization method on mini-batch for computing fixed statistics), mirror sampling (sampling a pair ($-\epsilon$, ϵ) for estimation), and fitness shaping techniques.

4. Experimental Results and Analysis

4.1. Data and Evaluation Indexes

Binance is the world's largest cryptocurrency exchange. The data in this article comes from Binance Public Data (https://github.com/binance/binance-public-data), an official open-source project published on GitHub by Cryptocurrency. Developers are free to download the K-line data, transaction data, and aggregated data. We download the K-line data, *i.e.* OHLCV data.

From the period of the data, Bitcoin started to generate the quotation data only after the spot market of Coinan.com was launched in late 2017. Based on the special nature of digital currency market transactions and the volatility of the global economy, the K-image data for the whole year of 2021 was chosen as the training set, and the test set was chosen for January, February, September, and November of 2022. In terms of market trends, the above data include sharp up, sharp down, shock up, shock down, and continuous shock, which can comprehensively examine an agent's decision-making performance. In terms of trading frequency, we use 1-minute K-line data. The reduction of time granularity makes the market more diverse, and with more trading opportunities, which requires a higher level of agent trading decisions. The agent observes the market and makes a decision once a day to every 1-minute frequency, in terms of the daily line for a large up or down trend market, in terms of the 1-minute line, more performance for the oscillator market.

In this paper, we use cumulative log-returns (CLR), maximum retracement (drawdown) (MDD), and rate of return (RR) to measure the return capability and risk tolerance of different agent models in reinforcement learning.

4.1. Experiments and Results Analysis

Table 2 shows the comparison tests of the Q-learning agent, evolutionary strategies agent, and policy gradient agent on the test dataset for each month. From the table, it can be seen that the agent model with evolutionary strategies (ES) achieves the best results in terms of return. As shown in Figure 3–5, the best results were achieved in January-February, September, and November with 59.18%, 25.14%, and 22.72%, respectively, and the final convergence of the overall model was better than the other agent models.

The experimental results show that the ES strategy has better decision performance in dealing with sharp rises, sharp falls, and shocks, demonstrating sufficient risk tolerance. In Figures 3-5, one can observe the characteristics of trading slippage exhibited by agents when confronted with sharply fluctuating K-line charts (rapid rises or falls) and volatile markets, highlighting the dual challenges of market liquidity and strategy adaptability. Specifically, in trending markets, ample liquidity, fueled by high trading volumes, enables agents to execute trades at prices close to their expected levels even amidst rapidly moving prices, resulting in relatively sparse slippage. This is not only attributed to the favorable conditions offered by the market itself but also to the agent's capability to timely discern and align with trends, implementing efficient order execution strategies.

Table 2. Yield Comparison.

model	Jan - Feb	Sept	Nov
Q-learning agent	1.01%	0.07%	0.53%
ES agent	59.18%	25.14%	22.72%
PG agent	–74.40%	–90.74%	–95.83%
BTC/USD	6.86%	–3.23%	–19.35%

ES: evolutionary strategies, PG: policy gradient.

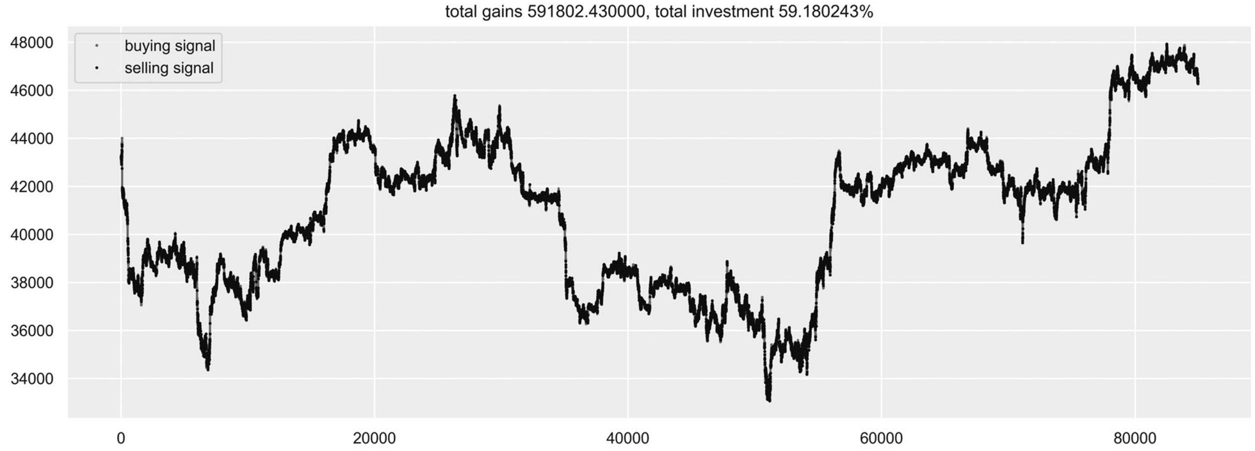

Figure 3. January-February ES model trading revenue graph.

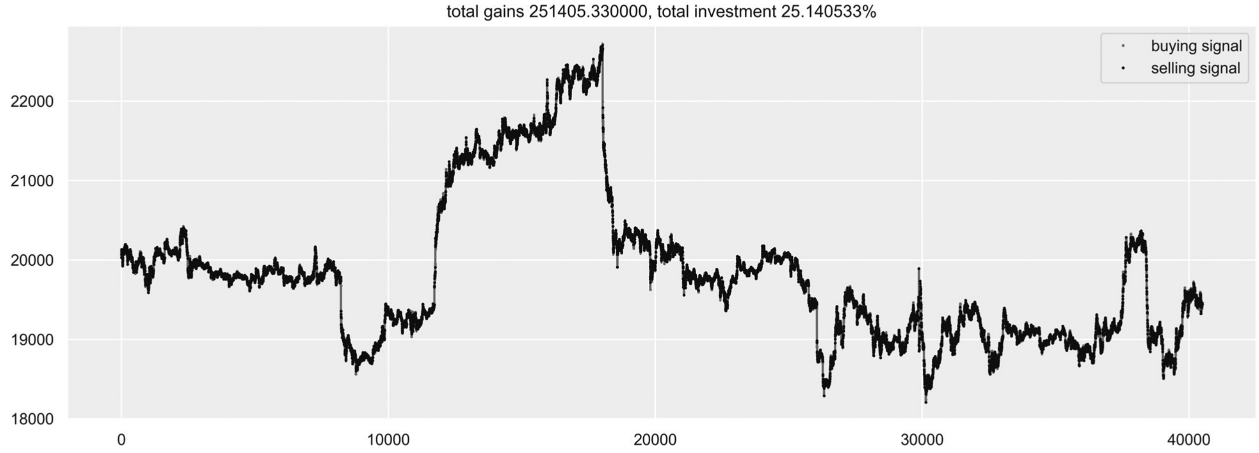

Figure 4. September ES Model Trading Returns graph.

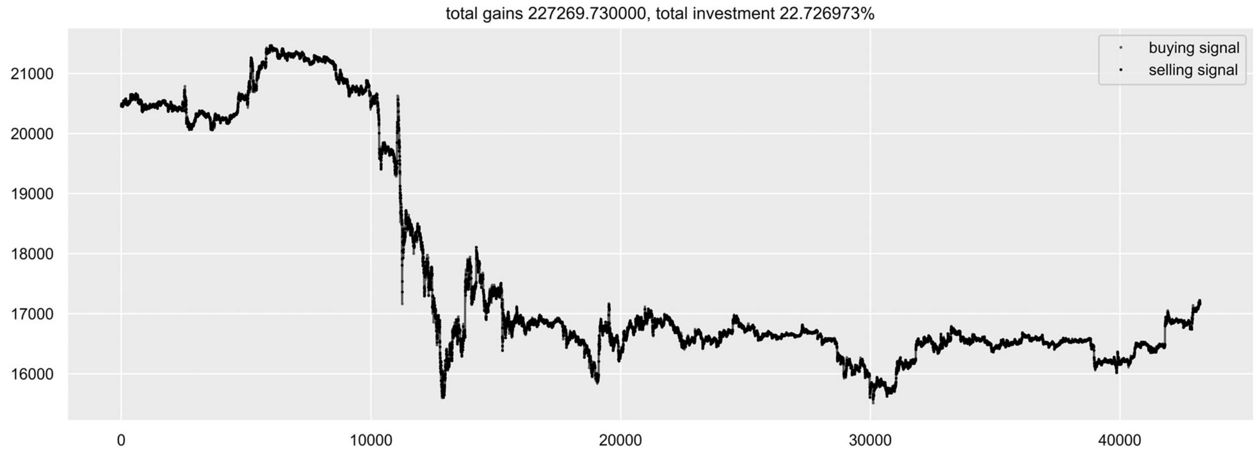

Figure 5. November ES Model Trading Returns graph.

In contrast, in a volatile market, due to frequent and limited price fluctuations and insufficient market depth, it is easier for intelligent agents to encounter increased slippage, especially when using less flexible trading strategies. This phenomenon prompts intelligent agents to have more complex and nuanced market interpretation abilities, as well as powerful learning and adaptation mechanisms, in order to optimize order types, adjust risk parameters, and even prudently reduce trading activities during high uncertainty. This serves as an effective means to reduce slippage and protect capital. Therefore, by understanding and responding to these market characteristics, intelligent agents can not only reduce unnecessary trading costs but also enhance their profitability and strengthen their survival and competitiveness in a dynamic market environment. This demonstrates a high level of flexibility and strategic optimization advantages.

From the analysis of the logarithmic return and the extent of the maximum retracement depicted in Figure 6, the ES model exhibits a remarkable capacity for generating favorable returns. This observation underscores the ES model's robustness when confronted with a substantial volume of high-frequency data, a testament to its versatility and broad applicability in the field of global optimization methodologies. Characterized by self-organization, self-adaptation, and self-learning capabilities, the ES model transcends the confines of specific problem domains, thereby demonstrating an inherent flexibility to address a wide array of challenges. Its unique selling points lie in its ability to autonomously fine-tune its strategies in response to dynamic market conditions and learn from past interactions, thereby enhancing its performance iteratively. This self-evolving mechanism empowers the ES model to distill crucial, underlying patterns from the intricate interplay between transaction volumes and timing – facets of trading data that often elude conventional optimization algorithms due to their complexity and multidimensionality. Consequently, the ES model emerges as a potent tool for navigating through the labyrinth of complex financial landscapes, where it excels at unraveling and capitalizing on subtle trends and patterns that would otherwise remain obscured.

5. Conclusion

Without specifying trading rules, the agent of the model proposed in this paper is able to rely on his exploration and utilization, trial and error in the training environment, and finally, through the learning experience, is able to master the trading skills and achieve the set goals, showing a better level of trading decisions – grasping the market trend. The company has been able to grasp the market trend, to judge the trend more accurately, to output trading decisions, to effectively reduce the maximum retracement, to improve the rate of return and risk resistance, and to make the trading process more robust.

In this process, the evolutionary algorithm plays a major role in processing the market state. Using the evolutionary algorithm to process the factor data is superior to using the default parameters

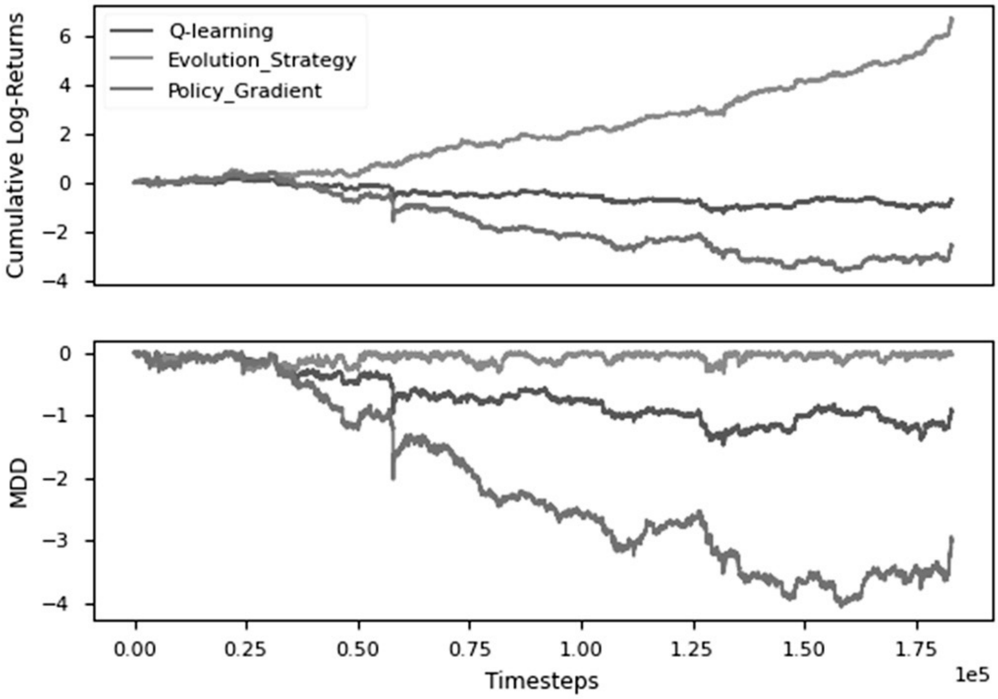

Figure 6. Cumulative log-returns and maximum retracement graph.

to calculate the factor data and it outperforms the DQN network using the Q-learning algorithm and the DNN network using the policy gradient algorithm as the agent output decision.

However, further research is needed to explore the scalability and generalizability of the approach to other digital currencies and trading scenarios. Future research directions include incorporating additional market data sources, investigating the integration of domain knowledge into the learning process, and exploring the application of the DERL model to other financial markets. Moreover, the development of interpretable reinforcement learning methods could provide valuable insights into the model's decision-making process and enhance its transparency.

Acknowledgement

This study was funded by Special Project in Key Fields of Colleges and Universities in Guangdong Province, China (grant number NO.2021ZDZX3006), and Guangzhou Science and Technology Plan Project, China (grant number NO.202201011651).

References

[1] L. Han *et al.*, "Efficient Continuous Space Policy Optimization for High-frequency Trading", in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023. http://dx.doi.org/10.1145/3580305.3599813

[2] G. Mattera and R. Mattera, "Shrinkage Estimation with Reinforcement Learning of Large Variance Matrices for Portfolio Selection", *Intelligent Systems with Applications*, vol. 17, 2023. http://dx.doi.org/10.1016/j.iswa.2023.200181

[3] X. Jie *et al.*, "Research on Financial Trading Algorithm Based on Deep Reinforcement Learning", *Computer Engineering and Applications Journal*, vol. 58, no. 7, pp. 276–285, 2022. http://dx.doi.org/10.3778/j.issn.1002-8331.2109-0507

[4] F. Liu *et al.*, "Bitcoin Transaction Strategy Construction Based on Deep Reinforcement Learning", *Applied Soft Computing*, vol. 113, 2021. http://dx.doi.org/10.1016/j.asoc.2021.107952

[5] Z. Duan *et al.*, "Optimal Action Space Search: An Effective Deep Reinforcement Learning Method for Algorithmic Trading", in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 406–415. http://dx.doi.org/10.1145/3511808.3557412

[6] J. Li *et al.*, "Online Portfolio Management Via Deep Reinforcement Learning with High-frequency Data", *Information Processing & Management*, vol. 60, no. 3, 2023. http://dx.doi.org/10.1016/j.ipm.2022.103247

[7] G. Chung *et al.*, "Market Making under Order Stacking Framework: A Deep Reinforcement Learning Approach", in *Proceedings of the 3rd ACM International Conference on AI in Finance*, 2022, pp. 223–231. http://dx.doi.org/10.1145/3533271.3561789

[8] X. Chen and H. Guo, "A Futures Quantitative Trading Strategy Based on a Deep Reinforcement Learning Algorithm", in *Proc. of the 2023 IEEE 8th International Conference on Big Data Analytics (ICBDA), Harbin, China*, 2023, pp. 175–179. http://dx.doi.org/10.1109/ICBDA57405.2023.10104902

[9] A. Alameer and K. Al Shehri, "Conditional Value-at-Risk for Quantitative Trading: A Direct Reinforcement Learning Approach", in *Proc. of the 2022 IEEE Conference on Control Technology and Applications (CCTA), Trieste, Italy*, 2022, pp. 1208–1213. http://dx.doi.org/10.1109/CCTA49430.2022.9966017

[10] T. Cui *et al.*, "Portfolio Constructions in Cryptocurrency Market: A CVaR-based Deep Reinforcement Learning Approach", *Economic Modelling*, vol. 119, 2023. http://dx.doi.org/10.1016/j.econmod.2022.106078

[11] U. Suliman *et al.*, "Cryptocurrency Trading Agent Using Deep Reinforcement Learning", in *Proc. of the 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), Toronto, ON, Canada*, 2022, pp. 6–10. http://dx.doi.org/10.1109/ISCMI56532.2022.10068485

[12] J. Wang *et al.*, "Stock Trading Strategy of Reinforcement Learning Driven by Turning Point Classification", *Neural Processing Letters*, vol. 55, no. 3, pp. 3489–3508, 2023. http://dx.doi.org/10.1007/s11063-022-11019-w

[13] Z. Huang *et al.*, "TBDQN: A Novel Two-branch Deep Q-network for Crude Oil and Natural Gas Futures Trading", *Applied Energy*, vol. 347, 2023. http://dx.doi.org/10.1016/j.apenergy.2023.121321

[14] P. Zhao *et al.*, "Improving Quantitative Stock Trading Prediction Based on MAD Using Q-learning Technology", in *Proc. of the 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India*, 2023, pp. 57–60. http://dx.doi.org/10.1109/AISC56616.2023.10085634

[15] W. Wei *et al.*, "Bitcoin Transaction Forecasting With Deep Network Representation Learning", in *Proc. of the IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1359–1371, 2021. http://dx.doi.org/10.1109/TEtc.2020.3010464

[16] C. Zhang *et al.*, "DoubleEnsemble: A New Ensemble Method Based on Sample Reweighting and Feature Selection for Financial Data Analysis", in *Proc. of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy*, 2020, pp. 781–790. http://dx.doi.org/10.1109/ICDM50108.2020.00087

[17] R. Liu and J. Zou, "The Effects of Memory Replay in Reinforcement Learning", in *Proc. of the 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA*, 2018, pp. 478–485. http://dx.doi.org/10.1109/ALLERTON.2018.8636075

[18] D. E. Moriarty *et al.*, "Evolutionary Algorithms for Reinforcement Learning", *Journal of Artificial Intelligence Research*, vol. 11, pp. 241–276, 1999. http://dx.doi.org/10.1613/jair.613

[19] T. Salimans *et al.*, "Evolution Strategies as a Scalable Alternative to Reinforcement Learning", arXiv preprint arXiv:1703.03864 (2017). http://dx.doi.org/10.48550/arXiv.1703.03864

Received: April 2024
Revised: June 2024
Accepted: June 2024

Contact addresses:
Yijun He
School of Information Science
Guangdong University of Finance and Economics
China
Guangdong Intelligent Business Engineering Technology
Research Center
China
e-mail: 980829798@qq.com

Bo Xu*
School of Information Science
Guangdong University of Finance and Economics
China
Guangdong Intelligent Business Engineering Technology
Research Center
China
e-mail: xubo807127940@163.com
*Corresponding author

Xinpu Su
School of Information Science
Guangdong University of Finance and Economics
China
Guangdong Intelligent Business Engineering Technology
Research Center
China
e-mail: 326918182@qq.com

Yijun He received the B.Eng. degree from the School of Computer Science & Engineering, TIANHE College of Guangdong Polytechnic Normal University, China, in 2021. He is currently a master student in the School of Information Science, Guangdong University of Finance and Economics, China. His current research interests are in artificial intelligence methods for quantitative trading.

Bo Xu received the B.Eng. degree from Hengyang Normal University, China, in 2005, MSc degree in computer science and technology from the Hunan University, China, in 2009, and PhD degree in software engineering from the South China University of Technology, China, in 2017. He is currently an associate professor in the School of Information Science, Guangdong University of Finance and Economics, China. His research interests include machine learning and intelligent optimization algorithms.

Xinpu Su received the B.Eng. degree from the School of Information Science, Guangdong University of Finance and Economics, China, in 2021 and M.Eng. degree from the School of Information Science, Guangdong University of Finance and Economics, China, in 2024. His current research interests are in artificial intelligence methods for quantitative trading.